JHD's Notes on ICMS 2018

Notes by James H. Davenport

24–27 July 2018
JHD only 24/morning of 25

Contents

Chapter 1

Stein: COCALC: Making Open Source Mathematical Software Available on the Web

Collaborative Calculations in the Cloud. Demo

1. Various document types: he chose Jupyter notebook.
2. Time travel" lets one look at versions of the document.
3. Also SAGE worksheets, which are more like a single document.
4. Inside his LaTeX environment, there's a `\sage` command.
5. Can use "manage a course" features.

Looks like a theorem — easy to read but hard to write

Q Own Python library?

A Yes (but couldn't demo this). We have Ubuntu Linux as the base.

Q File rights?

A Via projects and allowing people to access these via projects. Currently 600+K of these.

Q Privacy?

1.1 Architecture

1. Monolith. We use this for developing COCALC itself.
2. Small number of versions, one account/project.

First implementation was via standard differences. Then switched to RethinkDB with Changefeeds. Eventually switched to Listen/Notify in Postgres, since RethinkDB failed. Replaces differential sync with a new one based on revision logs. We've fixed many subtle bugs over the years, but none known today. There's curiosity over the semantics of undo in shared documents!

1.2 Revision log algorithm

Need a database with Changefeeds. Meteor.js, Firebase (Google) does this. RethinkDB did a part of this, and Horizon.js was a new product from them. But they went bust.

Now Postgres. Open source, but tangled in COCALC code. A patch log is a sequence of triples (time, user, patch) with unique times. The current state of a document is the result of applying a sequence of patches in the right order (no merge conflicts!).

Clients synchronise clocks with a central server (to within a few seconds). When a client is editing is computes diffs and sends a stream of patches. If all clients stop editing, they'll converge to the same document.

The real problem is algorithms to do this efficiently. Combination of browser and backend, hence node.js is very helpful!

In 2006 started Sage notebook, inspired by Mathematica and Macsyma notebooks, and IPython command line. 2011 IPython became Jupyter. Very successful. But there's no backend state really: multiple clients opening the same notebook at once, or huge outputs etc. Three ways of Jupyter in COCALC.

1. Classical with realtime sync, and this was a nightmare with version changes of Jupyter.
2. Rewrite with React.js, so backend has full knowledge. Took three months in2017. Doesn't work with the extensions, since the API is different.
3. "Plain Classical Jupyter".

Sage worksheets has the performance problems of hundreds of editors on the same page, so changed this.

Chatrooms can run on the side of *any* file.

1.3 Commercial or Academic

SAGE and Jupyter are not-for-profit. COCALC is a commercial project. Why? *sagenb.org* was attacked by hackers, had malware, periodically problems with network, and Legal problems with university liability.

Hence became commercial. Grants don't really reduce my teaching much. I've been on unpaid leave for two years (my department chair does a lot of Sage development).

Chapter 2

Computational Algebraic Geometry

2.1 Xiaoxian Tang — Investigating multistationarity in structured reaction networks

Alicia Dickenstein, Mercedes Perez Millan, Anne Shiu, and and speaker.

Reaction M+MAPKK$\leftrightarrow \cdots$. Six species concentrations x_i and rate constants k_i. Note the conservation laws. Definition of a steady state. For given k, are there multiple stationary states? Can rename to six equations with six constants. Have $K_i = x_6/x_5$ and two more, which we can plug into Jacobian. Get a critical function $C(x) := \cdots$.

Theorem 1 *Suppose G admits a positive parametrisation and let G be a conservative network. Assume steady states but no boundary steady states. Let $r = s - \operatorname{rank}(W)$ where W in the conservation law matrix.*

- *If* $\operatorname{sign}(C(x^*)) = (-1)^r$ *for all* $x^* \in \mathbf{R}^s_{\geq 0}$ *then there is exactly one steady state.*
- *??*

In our simple example, this proves no multistationarity.

Bigger network with nine x_i. The conditions say there can be multistationarity, but we need to find a witness. Critical functions here also have the form $\frac{x^\alpha}{x_\beta} \sum m_i$ where the m_i are squarefree and homogeneous. This is in fact a general result, if the steady states are given by binomials. See ISSAC 2018.

Bigger example with 17. This one has a boundary steady state with many $x_i = 0$.

2.2 Luis David Garcia Puente - Neural Ideals in SageMath

Ethan Petersen, Nora Youngs, Ryan Kruse, Dane Miyata, Rebecca Garcia and speaker. There's a GitHub project on this.

1970s was a breakthrough in neural science: Place cells are active depending on where you are. So a variety may be active, as place cells overlap. This gives a code work for the firing at any place. A set of codewords is convex if it can be drawn in some $\mathbf{R}^d$ with convex sets.

Define neural ideal

$$J_C := \left\langle \prod_{i\in\sigma} x_i \prod_{j\notin\sigma} (1-x_j) | \sigma \notin C, \sigma \subset [n] \right\rangle$$

Every relation between the place fields is encoded by a polynomial in this ideal.

There are two canonical forms for such ideals $CF^i(C)$. If C is closed under intersections then $CF^2(C)$ has only short pseudomonomials $(prod_{i\in\sigma} x_i)\,(1-x_j)$ and in this case C is convex. $[n] \in C \Leftrightarrow CF^1(C) = \emptyset$, and in this case C is convex.

Use primary decomposition and other tools. Algorithms 1 is an iterative step to update $CF^1(C)$.

2.2.1 CF versus GB

Theorem 2 *If the canonical form is a GB for some ordering, it's actually a universal GB.*

There's a theorem giving various types of ideals

Q How big can you handle?

A Can do 20.

2.3 Hashemi et al. - Universal Groebner Basis for Parametric Polynomial Ideals

Given by JHD.

Q–TS Note the importance of [Wei87]. Also, can we get a *universal comprehensive GB*, rather than just a GS?

A Need to check with authors.

2.4 Hong et al. - Global Identifiability of Differential Models

Phosphorylation. We can measure B and C, but concentrations of others can't. Can we determine all the rate constants from such measurements. $\dot{x} = x + k$ is clearly identifiable, but $\dot{x} = x + k_1 + k_2$ clearly is not, only $k_1 + k_2$. Example: predator/prey when we can only measure prey. It turns out that k_2 is not identifiable, but the others are. Note that failing to spot non-identifiability is a problem for experimental design.

There's also a problem of local identifiability, e.g. $k_1 = \pm\sqrt{\cdots}$. This actually has much better algorithms than global.

- Taylor series: termination only for special cases
- Differential elimination for parameters, tackles only small examples
- Input-Output equations can give incorrect answers for conservation law systems. [Ollivier1990].

We reduce this differential problem to algebraic.

Theorem 3 *k_1 is identifiable iff there is an integer M and polynomials P, Q with*

$$k_1 = \frac{P(x_1, \ldots, x_m, \dot{x}_1, \ldots, \dot{x}_m, \ddot{x}_1, \ldots, x_m^{(M)})}{Q(\cdots)}$$

where $x_{1..m}$ are the measurable variables.

Four examples of timings against other systems.

We use GB for the algebra. Hence a Monte Carlo algorithm.

Q–TS "Monte Carlo" means only false negatives?

A Either direction. But we can re-run, whereas DAISY is based on an assumption that might be true or false.

2.5 Nida Obatake - The capacity for Hopf bifurcations in the fully distributive dual-site phosphorylation network

Fully-distributive dual-site phosphorylation network: 9 biochemical species and 11 rate constants. Can I choose initial values and rate constants that produce Hopf bifurcations? Hard. Consider the irreversible version. Use Mass action kinetics as in section 2.1. Steady states are with derivatives zero so a polynomial problem, interested in positive orthant solutions. Can make any vertex of the Newton polygon dominate. Had to fix three rate constants.

2.6 Jonathan Hauenstein - Certifying reality of projections

With Avinash Kulkarni, Emre C. Sertöz, and Samantha Sherman (Graduate Student at Sandia). Want $x \neq y \in \mathbf{C}$ such that $\text{null}(A(X)) \cap \text{null}(A(y))$ have a nontrivial common real vector. Certify realness via Newton-invariant sets.

2.6.1 α-certify

[HS12]

Q What precision did you use?

A All rational numbers.

2.7 Robert H. Lewis - Image Analysis: Identification of Objects via Polynomial Systems

Want to identify objects from sensor information. Use MAGMA and Maple. This is a classic problem in computer vision. Consider "six line" problem, where a building etc. is known by its sixlines.

The 2D case is simple: three points. In 3D we need four points Measure four distances. Multivariate Resultant [Dixon1908] but it only works in ideal situations. [KSY94] fixed these problems. The resultant is a factor of the determinant of a certain matrix, and I have extended this. Many people use GB here, but it doesn't work so often.

Apparently six equations in eight variables, but $x_1 = y_1 = x_2 = 0$ can be assumed in 2D. .02 seconds for me and several seconds for FGb in Maple. In 3D I can to in 10MB, but FGb takes GB and MAGMA never terminated.

With flexible wings, took 7 seconds and 32MB, but FGb took 9 minutes and 6.8 GB. Again, MAGMA didn't finish. The EDF (Early Detection of Factors) feature is crucial, because in fact the determinant is divisible by R^4.

2.8 Dan Bates, Danielle Brake, Matt Niemerg - Paramotopy: Parameter homotopies in parallel

How about `\beamernavigationsymbolsempty`!

More seriously, you don't know what's happening with homotopy: which start points will go to infinity. What happens with parametric systems? Hence Paratopy. Written on top of Bertini 1. C++ and you must compile (also Bertini 1). `www.paratopy.com`. Motivation is two robots in the same space.

This is better than just repeated sampling, because we have good parallelisation, and good transmission of sensible-sized batches back to the head node.

Chapter 3

General Session

3.1 Michael Monagan and Baris Tuncer: Sparse multivariate polynomial factorization: a high-performance design and implementation

In particular parallelise multivariate Hensel lifting. Notice that Wang's version works modulo some p^l, large enough to prove that we have the right result. This is only described in [GCL92] and is in Magma, Maple, Maxima (hence Sage), Mathematica and Singular.

Baris observed that, looking at the evaluations, in general the same monomials appear, and in fact there's an increasing chain of sets — *Strong Sparse Hensel Assumption*[1]. See [MT16]. Evaluate to $f_{j-1}(x_1, \gamma_m, \beta_3^k, \ldots, \beta_{j-1}^k)$. Hensel lift x_j. Dense interpolate x_2 then sparse evaluate remaining $x_3, \ldots$ using Vandermonde properties. With 16 cores he's seeing $\times 5$–$\times 10$ speedup.

3.1.1 Software

Using Cilk–C (MIT version), fork-join model. He uses arrays of machine (64-bit) integers as his representation. Arrays of coefficients and arrays of packed monomials. If necessary going to `__uint128_t` solves overflow problems. We use in place algorithms, as the memory-manager can be a bottleneck.

Q Have you ever seen the assumption violated?

A No, but I use 63-bit values. BT does with 31-bit.[2]

[1] Assuming merely that all monomials in coefficients of $x^l : l < n$ are monomials in coefficient of x^n would be the Weak assumption.

[2] JHD queried this. Of course he can make examples, in which case he just evaluates again. There's always a final check (probabilistic, as truly multiplying out would be big.

3.2 Annie Cuyt: SPARSIMATICS: a mathematics toolbox for sparse problems

With Engelbert Tijskens, Matteo Briani, Ferre Knaepkens and Wen-Shin Lee. Theory and applications. This toolbox is a byproduct of the advances made in the theory. Exponential analysis is basically an inverse problem.

Nyquist sampling isn't always possible. Consider sampling at $\tau + \sigma\Delta$ rather than Δ. Originally solved [dP95]. If $\sigma > 1$ we don't have uniqueness, also $\tau > 1$. However, if $\gcd(\sigma, \tau) = 1$ we are OK.

Example: 20-term exponential function from MRI. With $\sigma = 1$ we see a lot of clustering.With $\sigma = 11$, we are essentially powering the residues, and they separate. Hence this is much easier to count the multiplicity at least. Example of automotive radar. Again a great deal of cancellation.

Q-MM How do you choose σ?

A Some art, but the algorithm is also self-recovering.

3.3 Michela Ceria: Efficient computation of square-free separator polynomials

With Teo Mora and Andrea Visconti.

[CM90] given a finite set X of distinct points, compute the lexicographic Gröbner escalier $N(I(X))$. [FelszeghyRathRonyai2006] uses point-tries to compute with a better complexity. Not iterative on points. We use a trie saying that we have same path to level i if shape the first i coordinates.

So for one point, 1 is the polynomial. Then when I add a new point, I need to compute a new polynomial and update the old. For the new polynomial, at each level I need to distinguish it from all its siblings. Tested on a variety of examples in three/four coordinates. 65536 points in four coordinates took six minutes. Done over a finite field.

3.4 Business Session

3.4.1 AB: Chee Yap

In general, the previous general chair but one becomes the next AB Chair, but G.-M. Greuel can't, so JHD is taking over in advance. The general chair of ICMS 2020 will be Michael Joswig.

This is the sixth ICMS, and the community is still finding its way.

Note that we encourage you to propose your own special session at ICMS.

3.4.2 Local Organisers

AS said that JH had done most of the work. JH then spoke.

Notre Dame $\sim 25K$

Registration 100 full plus 25 students $\sim 24K$

NSF $20K.

Oak Ridge $4K

IMA direct support.

Travel Funding $37.5K — speakers plus travel support.

Banquet $20K in Studebaker Museum.

3.4.3 Programme Chair: MKauers

Sessions 20 accepted.

Contributions Talk or talk+paper. 59 proceedings papers and 43 talk only. The EasyChair organisation should be improved — we should have done everything through one system.

Length 3 or 4 parallel sessions? We went for 3, but we could have gone for 4 and slightly longer talks.

Calls Probably late (March 23) and should (MM commented) be end of previous year.

MKohlhase I missed having a demo track. General approval. Maybe a lighning talk + demo. General approval.

Proceedings These are memory stick versions of an LNCS volume. MM reported that an attempt to buy teh physical volume via the link on the website attracted a substantial (30%?) discount.

3.4.4 JHD for 2020

After ECM, and before ISSAC, therefore starting ≈14 July. Possibly co-located with (but after) ECM. If that doesn't work, co-located with (but before) ISSAC.

MM What about software tutorials etc.

3.4.5 AOB

Sperber Noted that this conference doesn't have guidelines on software citation.

JHD Noted for 2020 — to be in the call.

MJ And we should publish these practices in the proceedings as well.

TS Agreed but we need to check with Springer first.

Chapter 4

Hales: Formal Abstracts

Kepler conjectured that the cubic close packing (and hexagonal close packing) were optimal. Hales proof initially announced 9 August 1998. Definition of a formal proof, and list of some proof assistants. I actually use HOL-Light. Good real analysis was helpful. This project finished in 2014, which took a further three years to be refereed. There are $> 23,0000$ inequalities. 5000 processor hours for the verification in HOL-Light. Mostly on a 32-core machine, so elapsed time ~ 1 week.

See [KCS17].

FAbstracts is part of the IMU's GDML project.

> The relationship between the computer and mathematics is very different from that between the computer and the empirical sciences.

Given today's technology, it s not reasonable to ask for all proofs to be formalised. Nut with today's technology it seems that it should be possible to create a formal abstract service that

- Gives a statement of the main theorem(s) of each published mathematical paper in a language that is both human and machine readable
- Links each term in theorem statements to a precise definition of that term (again in a precise ...)
- ...

The two responses to Russell's paradox were Set theory (Zermelo) and Type Theory (Russell) [WR10]. Set theory moved into mathematics, and type theory into computer science. Need dependent types to allow for $3 \in \mathbf{N}$ and $3 \in \mathbf{R}$. Mizar is set theory, Isabelle agnostic, most of the rest are type theory. See [GT12], who has a "universal group" of which everything else is a subgroup.

Proof assistants have to deal with identification. Univalence is *an* answer to this.

Shows some Lean, a proof assistant and document editor.

Also looks at the Stacks project. Kevin Buzzard has been translating this into Lean. Shows `structure scheme :=` code.

Paradox 1 (Hypergame) *Some games finish, some don't. Hence Hypergame. First move: pick a game that necessarily terminates. The remaining moves are to play this. Therefore hypergame terminates. But then what happens if I pick hypergame.*

Get a bizarre error message from Lean when trying this. "universe level is too big".

Quotes also SEL4, CompCert as well as Feit–Thompson. OED has 273K words, medical corpus 600K. How many for mathematics? Quotes "normal" as an example of overloaded. Quantum groups are not groups, etc.

Quoted various powerful recent results, and gave a view of how much you would need to define.

Q–MK What do you think you need from subtypes?

A I don't have a formal list, but look at Gonthier's components project.

Q Why Lean?

A There's a personal reason: it's at CMU. It has a very similar foundation to Coq.

Q-MJ HTT?

A There is some Lean support for this.

Q *abc* conjecture?

A I understand that a lemma has been pinpointed as highly suspicious. So if we know where to look that might help.

Chapter 5

25 July Mathematical Knowledge Management

5.1 Katz: Software Citation: theory and Practice

SSI's survey [HAL+14], and my survey of the US National PDR association. very similar numbers. 2/3 "my research would not be possible without software". In *Nature*, about 6.5 software packages/article.

The citation system was developed for papers and books. We either force software to fit into this, or we invent a totally different system. Many think the second is too hard (but I'd love to see it happen), so I'll concentrate on the first. Note that we want to look at software throughout the research process, not just in the publication. Started a Software Citation group under the aegis of Force 11.

Six principles.

- First class
- Credit and Attribution
- Unique
- Persistence
- Accessibility to software and necessary materials
- Specificity (actual version used).

GitHub has a guide on how to make your software publishable (basically via Zenodo). This gets a DOI (or equivalent). Then create a `CITATION` file, and probably update the `README`. Note that you can also write a software paper.

To cite software, first check if they've told you. If not, have to follow the principles as best possible. Unless you know better, cite the authors as "The XXX project". Make sure that you cite a landing page (as DOIs do for papers).

Now a Software Citation Working Group.

Q

A

5.2 Hagen Chrapary and Wolfgang Dalitz – Software products, software versions, archiving of software, and swMATH

Introduction to zbMATH: 4M entries from 3000 journals and 180,00 books. About 7000 active reviewers.

Hence swMATH; collected from the zbMATH database but stored in our own database.

There are links to Internet Archive (wayback machine) and also to ?? where there is version control information (?).

Also links to SoftwareHeritage (a French initiative) which collects code from GitHub.

Growing usage statistics, but heavily influenced by robots.

Bibliography

[CM90] L. Cerlienco and M. Mureddu. Algoritmi combinatori per l'interpolazione polinomiale in dimensione ≥ 2. *Actes de la 24me Session du Seminaire Lotharingien de Combinatoire*, pages 39–76, 1990.

[dP95] G.C.F.M.R. de Prony. Essai expérimental et analytique sur les lois de la Dilatabilité des fluides élastique et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, á différentes températures. *J. de l'École Polytechnique*, 1:24–76, 1795.

[GCL92] K.O. Geddes, S.R. Czapor, and G. Labahn. Algorithms for Computer Algebra. *Kluwer*, 1992.

[GT12] G. Gonthier and L. Théry. Formal Proof — The Feit–Thompson Theorem. `http://www.msr-inria.inria.fr/events-news/feit-thompson-proved-in-coq`, 2012.

[HAL+14] S. Hettrick, M. Antonioletti, Carr L., N. Chue Hong, S. Crouch, D. De Roure, I. Emsley, C. Goble, A. Hay, D. Inupakutika, M. Jackson, A. Nenadic, T. Parkinson, M.I. Parsons, A. Pawlik, G. Peru, A. Proeme, J. Robinson, and S. Sufi. UK research software survey 2014. `https://doi.org/10.5281/zenodo.14809`, 2014.

[HS12] J.D. Hauenstein and F. Sottile. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Transactions on Mathematical Software pp. 28:1-28:20*, 38, 2012.

[KCS17] C. Kaliszyk, F. Chollet, and C. Szegedy. Holstep: A machine learning dataset for higher-order logic theorem proving. `http://arxiv.org/abs/1703.00426`, 2017.

[KSY94] D. Kapur, T. Saxena, and L. Yang. Algebraic and geometric reasoning using Dixon resultants. In *Proceedings ISSAC 1994*, pages 99–107, 1994.

[MT16] M. Monagan and B. Tuncer. Using Sparse Interpolation in Hensel Lifting. *International Workshop on Computer Algebra in Scientific Computing*, pages 381–400, 2016.

[Wei87] V. Weispfenning. Constructing universal Gröbner bases. In L. Huguet and A. Poli, editors, *Proceedings AAECC-5*, pages 408–417, 1987.

[WR10] A.N. Whitehead and B. Russell. Principia Mathematica. *Cambridge University Press*, 1910.